Deep reinforcement learning using sparse distributed memory

Michal Chovanec, michal.nand@gmail.com
Peter Šarafín, peter.sarafin@gmail.com

Keywords: reinforcement learning, function approximation, SARSA, Q-learning, sparse distributed memory, robotics, neural network
Abstract: In this paper we present advanced sparse distributed memory (ASDM) (based on original Kenerva's research [1] [2] [3]). We designed deep unsupervised features extraction algorithm combined with linear combination of features to store Q values of SARSA algorithm. On three experiments we show ability of to use sparse coding for Q-value approximation. Those experiments include ASDM topology experiments (including convolutional ASDM), noise imunity and learning speed in virtaul environment. **TODO** and finally - line following robot experiment.

1 Introduction

Reinforcement learning is well known method **TODO citovat Watkinsa** to solve Markovov decission problems **TODO citovat co to je**. Commonly used algorithms are Q-learning and SARSA (equation 1). This function is handling information about suitablity action $a(n)$ in state $s(n)$.

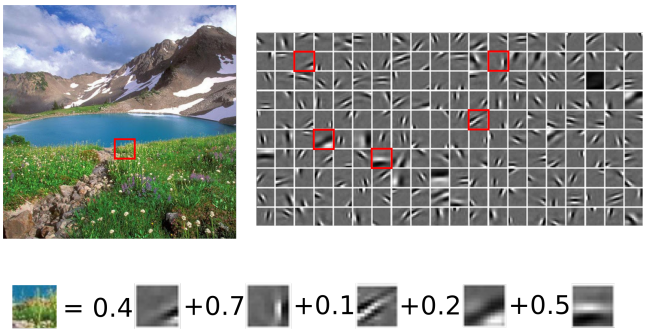
$$Q'(s,a) = (1-\alpha)Q(s,a) + \alpha\Big(R(s,a) + \gamma Q(s',a')\Big) \tag{1}$$

For small state spaces (usually only in examples) a table can be used to store Q values. For large state spaces - with hundreds of state features, the approximation is necessary. Well known are linear combinations of basis functions **TODO citovat** or feed forward neural networks **citovat**.

Reccurent character of equation 1 makes learning process inefficient, and slow.

Our idea is focused on linear combination of learned features. Deep features extraction part is learned unsupervised. On last stage is processed linear combination of features, and this part is learned supervised. Because of linear character, this part is learning fast. Another benefit from features extraction leads to build learning system with ability to use past knowledges to make new conclusion in situation which was never observed before, but have similar features like some past situation.

Those features should be sparse, reasons for sparsness are explained in well cited paper such [4], [5], and in medical papers focused on research of primary and visual cortex [6], [7], [8], [9]. Sparse coding uses linear combination of few selected words from dictionary to represents input, this shows image 1.



Obr. 1: Sparse representation principle

2 ASDM algorithm

Based on K-SVD algorithm **TODO citovat**. Optimization problem formalisation :

$$E(w,y) = \sum_{k=0}^{K-1} \|x_k - wy_k\|_2^2 + \lambda_0 \|y_k\|_1^2 + \lambda_1 \|w\|_{2,1}^2$$
$$\text{subject to} \quad y_{ki} > 0$$

- reconstruction condition

$$\|x_k - wy_k\|_2^2$$

- y_k sparsity condition

$$\lambda_0 \|y_k\|_1^2$$

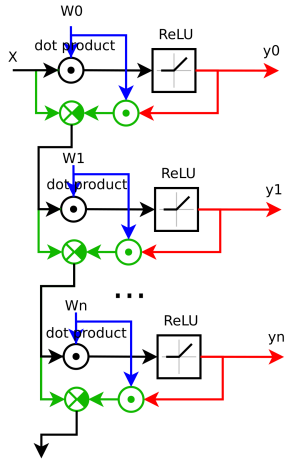
- w sparsity condition

$$\lambda_1 \|w\|_{2,1}^2$$

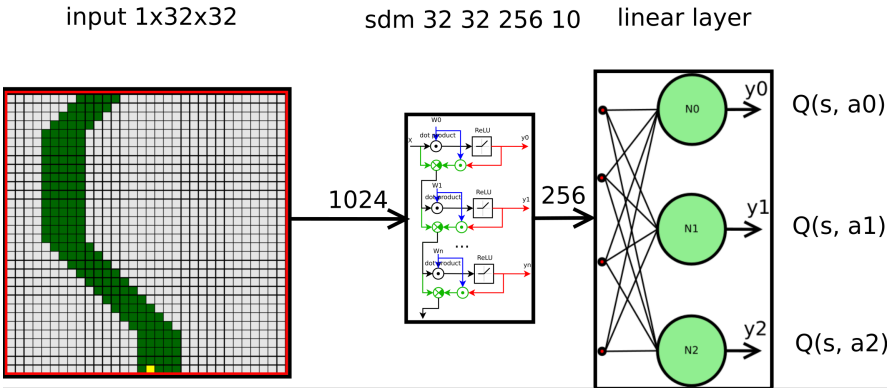
Our goal is to approximate function $y = f(x)$, where $x, y \in R$ are vectors. The x is input with size N, and have usually huge dimension (f.e. ten's of sensors, hundreds of camera pixels ...). Output y with size M can be class in classification problem, or any required value to approximate, usually with significantly lower dimension than x.

Learning process is done in two steps :
unsupervised learning, where Oja's rule is used to estimate input weights
supervised learning, where gradient descent is used to estimate output weights

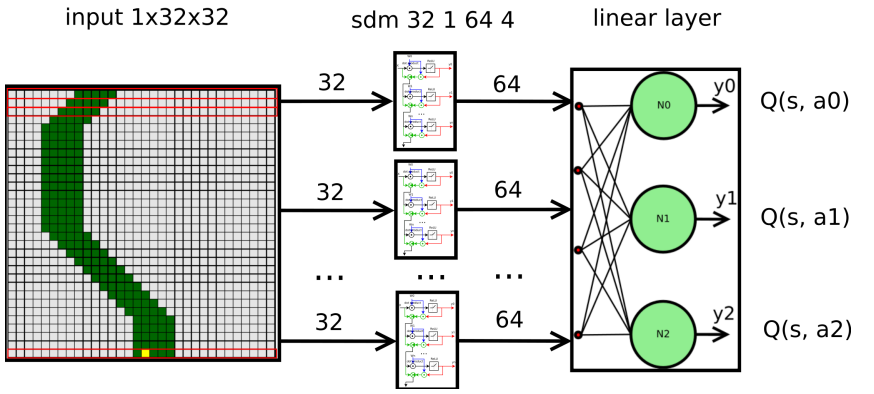
Data: W, depth, X
Result: sparse representation Y, features matrix weights W
initialization
$W \leftarrow random$, $R \leftarrow X$
η learning rate, ϵ shrink value
for *j from 0 to depth* **do**
  $best = \arg\max_i W[i]R$
  $d = ReLU(\frac{W[best]R}{\|W[best]\|_2^2})$
  $Y[best] = d$
  for *i from 0 to size(R)* **do**
    $dw = d(R[i] - W[best][i])$
    $W[best][i] = W[best][i] + \eta dw$
    $W[best][i] = shrink(W[best][i], \epsilon)$
    R[i] = R[i] - d*W[best][i]
  end
end



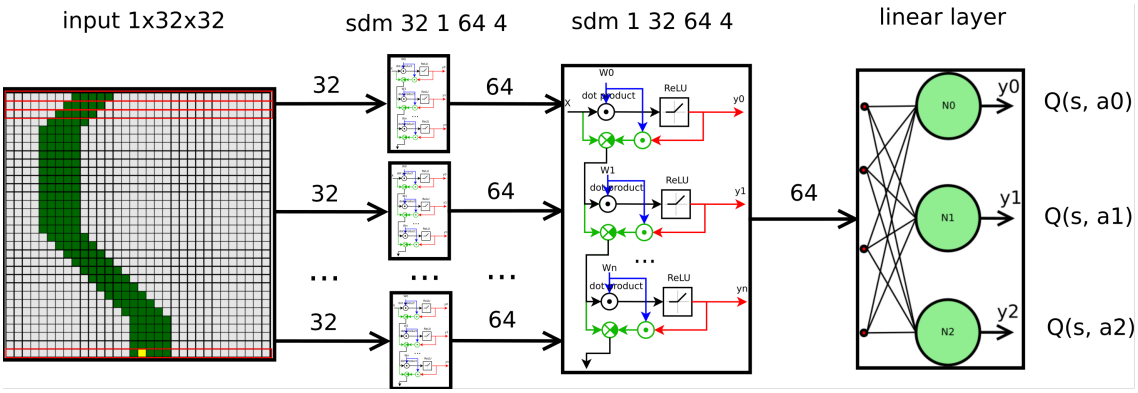
Obr. 2: Algorithm working princple



Obr. 3: Basic ASDM topology



Obr. 4: ASDM + Convolution topology



Obr. 5: ASDM + Convolution topology + Deep layer

3 Experimental results

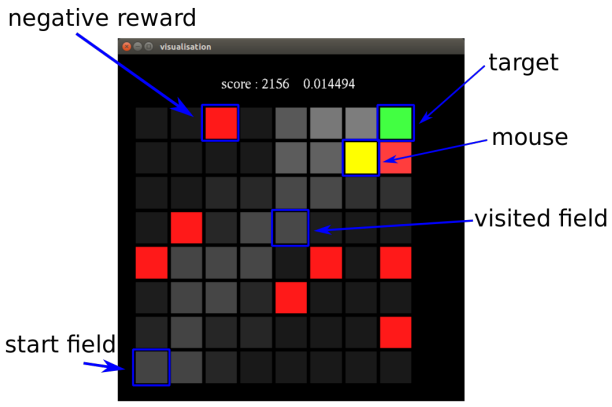
4 Mouse and food

- state : vector $64x\{0,1\}$, 1 if mouse on field
- action : left, right, up, down
- reward : +1 for target hit, -1 for red hit

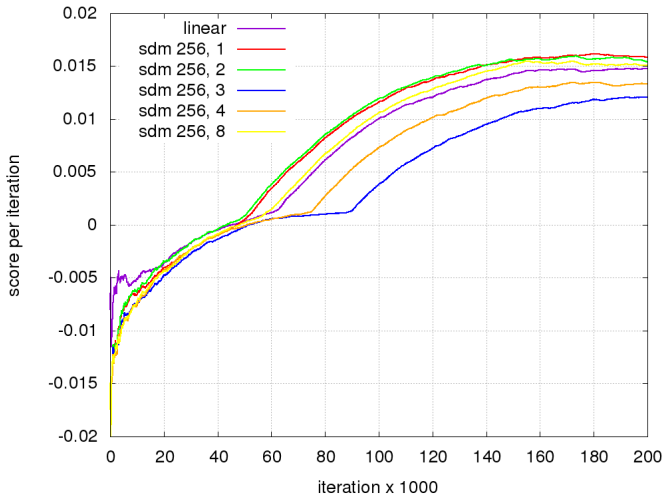
5 Virtual line follower

- motion equations of robot

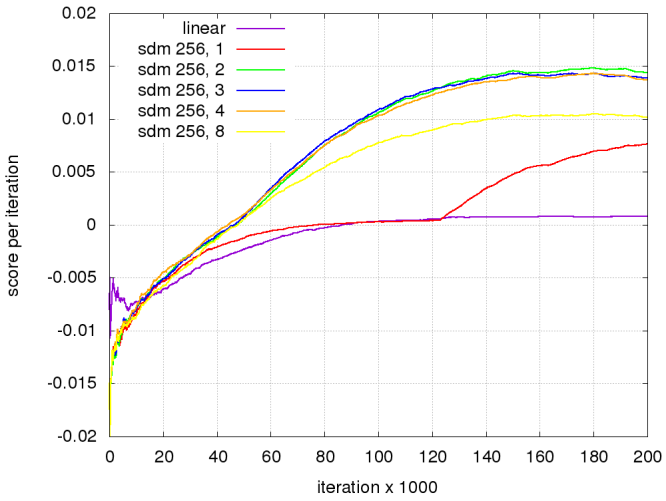
$$a(n) = \{-1, 0, 1\}$$
$$v(n) = \alpha r(n-1) + (1.0 - \alpha)a(n)$$
$$p(n) = p(n-1) + v(n)$$



Obr. 6: Mouse experiment description

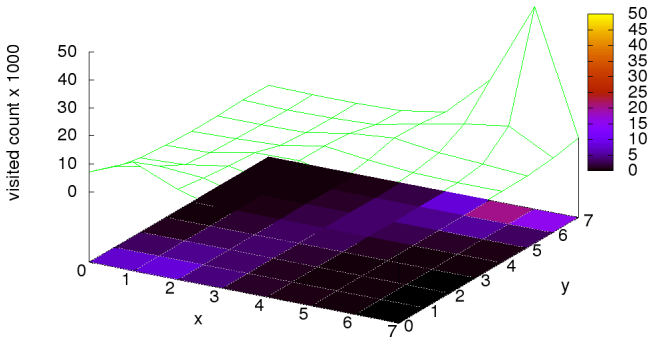


Obr. 7: Score progress rate, for state noise 0.0

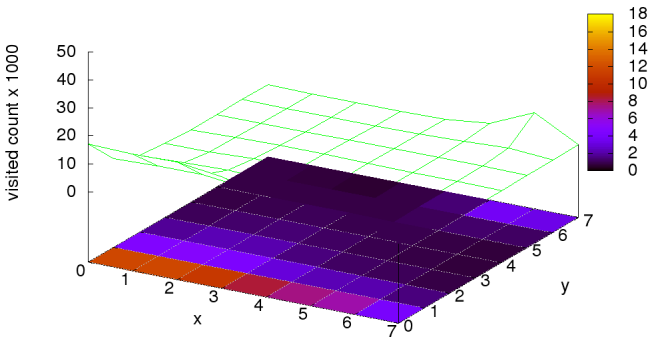


Obr. 8: Score progress rate, for state noise 0.1

visited fields rate for linear approximator



Obr. 9: Visited field rate, state noise 0.0



Obr. 10: Visited field rate, state noise 0.1

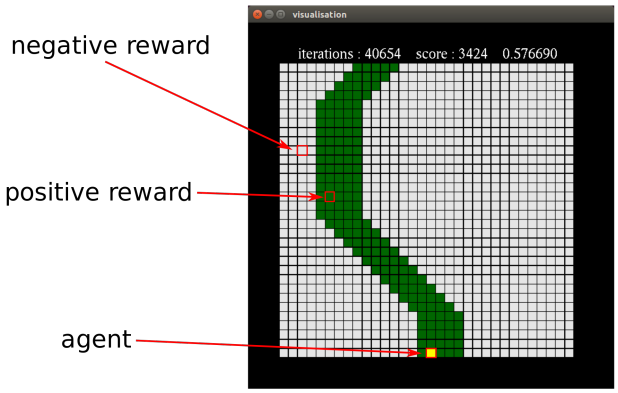
- reward

$$r(n) = \begin{cases} +1, & \text{if green field} \\ -1, & \text{otherwise} \end{cases}$$

6 Worms

Obr. 11: Visited field rate, state noise 0.0

Obr. 12: Visited field rate, state noise 0.1

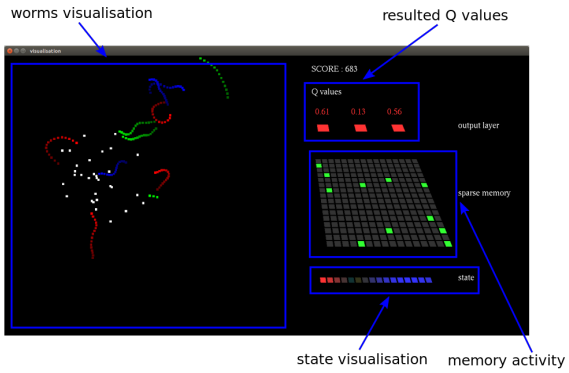


Obr. 13: Virtual line follower description

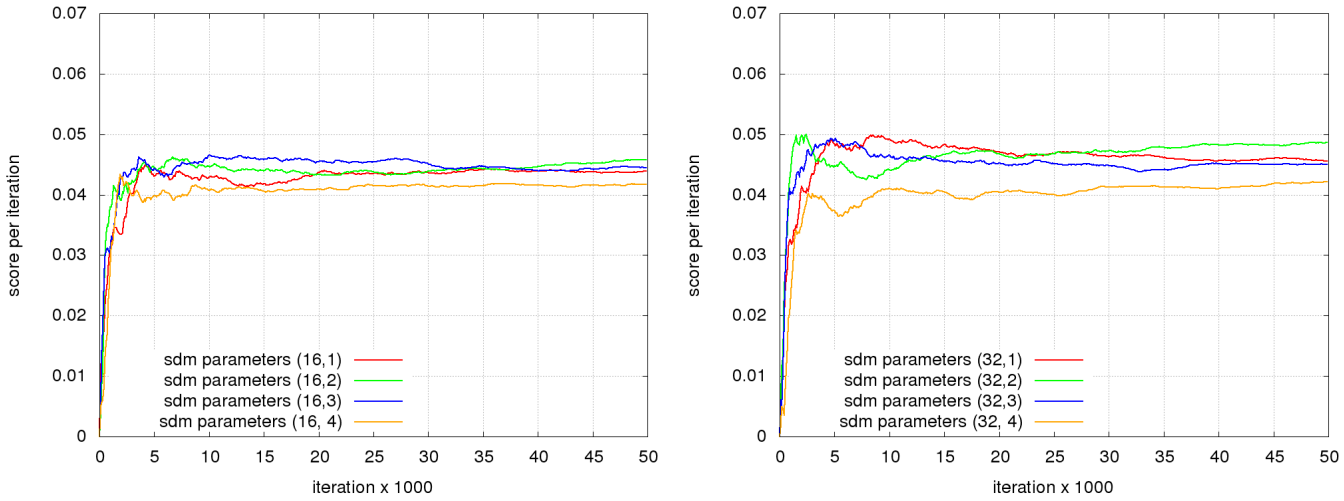
Obr. 14: Total score

Obr. 15: Progress per iteration

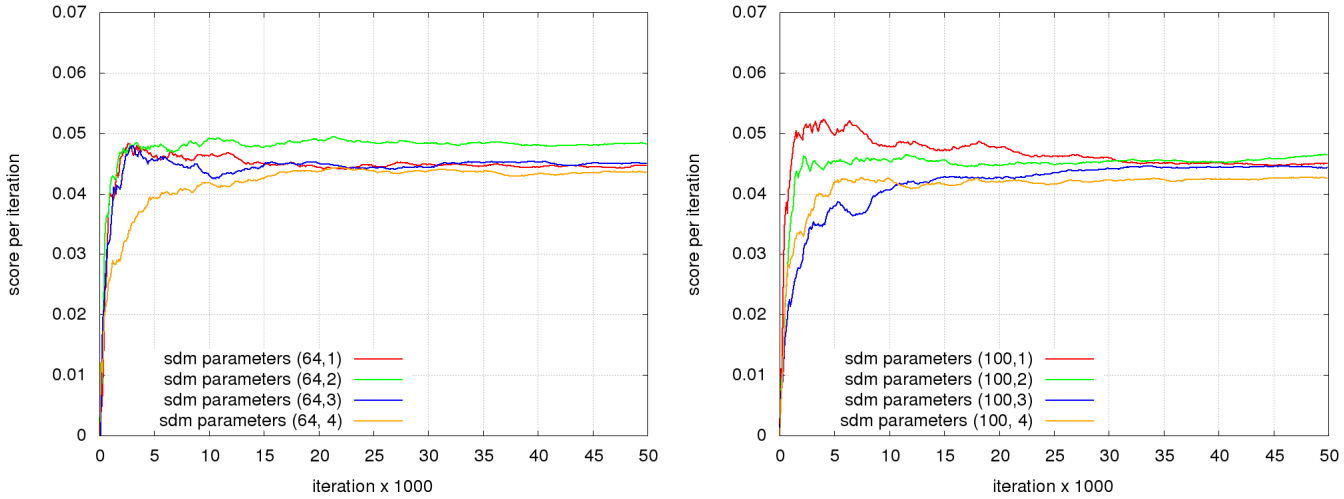
6.1 Conclusion



Obr. 16: Multiple worms testing application



Obr. 17: Memory size and activity results



Obr. 18: Memory size and activity results

Literatúra

[1] M.J.Flynn, P.Kanerva, and N.Bhadkamkar, 1989, Sparse Distributed Memory: Principles and Operation http://i.stanford.edu/pub/cstr/reports/csl/tr/89/400/CSL-TR-89-400.pdf

[2] David Rogers, 1988, NASA, KANERVA'S SPARSE DISTRIBUTED MEMORY: AN ASSOCIATIVE MEMORY ALGORITHM WELL SUITED TO THE CONNECTION MACHINE https://pdfs.semanticscholar.org/9288/bb551f000348f800ff40d0fdb3fd74c410ef.pdf

[3] J. S. Albus, 1975, Data Storage in Cerebellar Model Articulation Controller https://www.cs.cmu.edu/afs/cs/academic/class/15883-f13/readings/albus-1975.pdf

[4] Olshausen and Field (1997): Sparse coding with an overcomplete basis set

[5] Olshausen BA, Field DJ (2004) : Sparse coding of sensory inputs. Current Opinion in Neurobiology, 14, 481-487

[6] Mushroom body, locust (Laurent)

[7] HVC, zebra finch (Fee)

[8] Auditory cortex, mouse (DeWeese & Zador)

[9] Hippocampus, ratprimate(Thompson & Best; Skaggs)

[10] Motor cortex, rabbit (Swadlow)

[11] Visual cortex, monkeycat (Vinje & Gallant)

[12] Inferotemporal cortex, human (Fried & Koch)